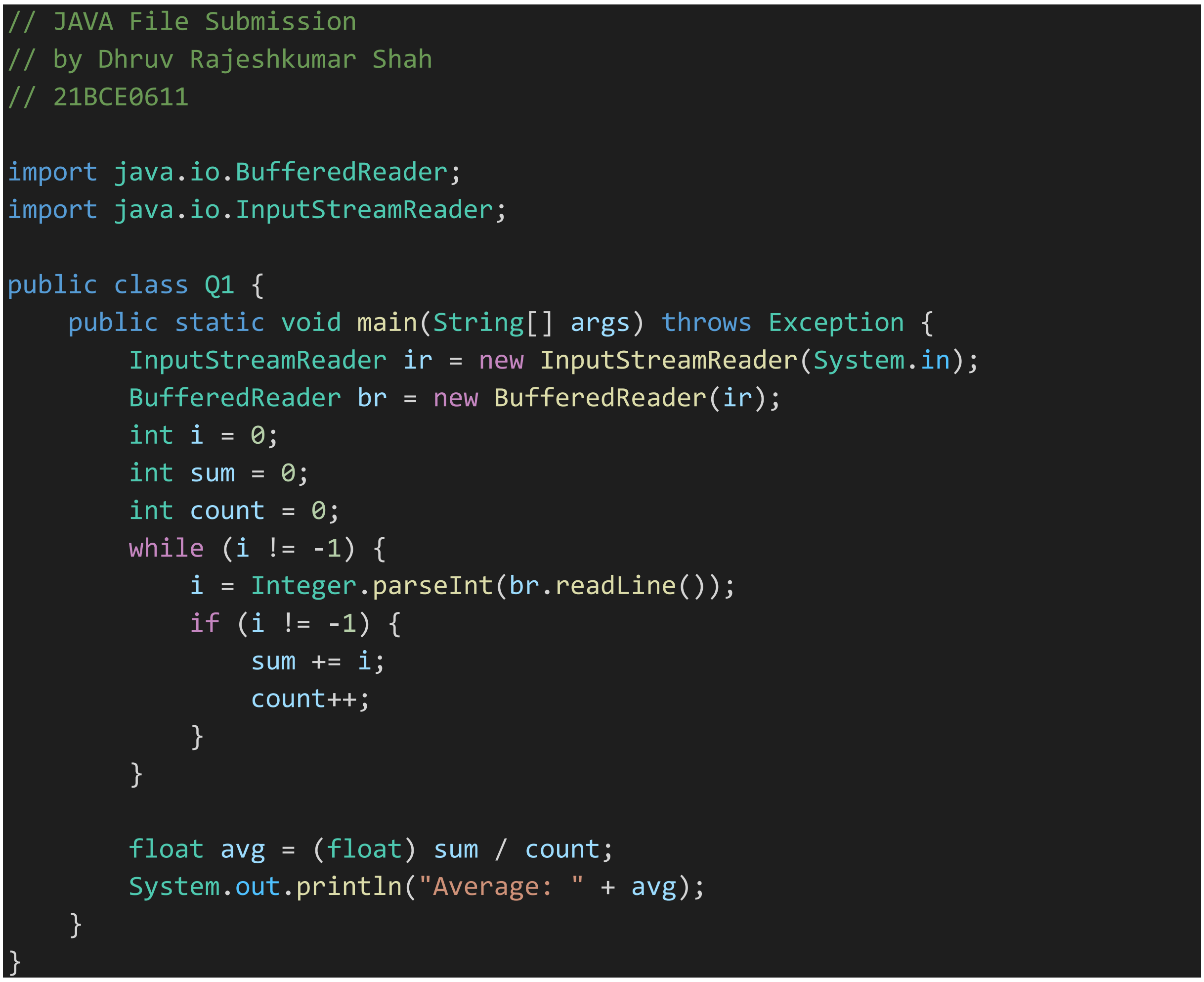
BCSE103E - Computer Programming: Java

File Submission

Name: Dhruv Rajeshkumar Shah
Registration No – 21BCE0611

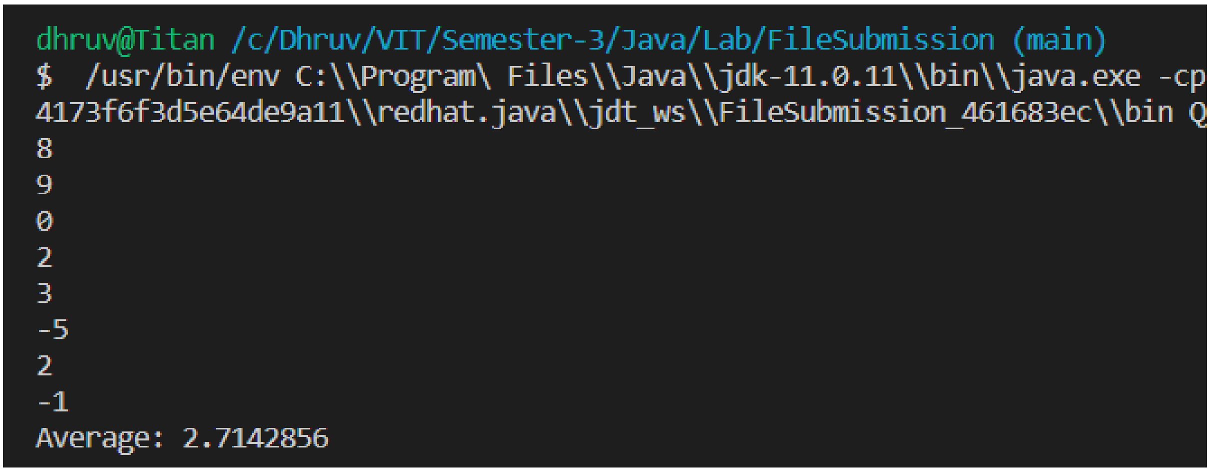
Question 1: => Create a buffered reader that reads integers separated by newline and terminated by -1. Find the average of all these numbers (except -1).

CODE



```
// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Q1 {
    public static void main(String[] args) throws Exception {
        InputStreamReader ir = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(ir);
        int i = 0;
        int sum = 0;
        int count = 0;
        while (i != -1) {
            i = Integer.parseInt(br.readLine());
            if (i != -1) {
                sum += i;
                count++;
            }
        }

        float avg = (float) sum / count;
        System.out.println("Average: " + avg);
    }
}
```

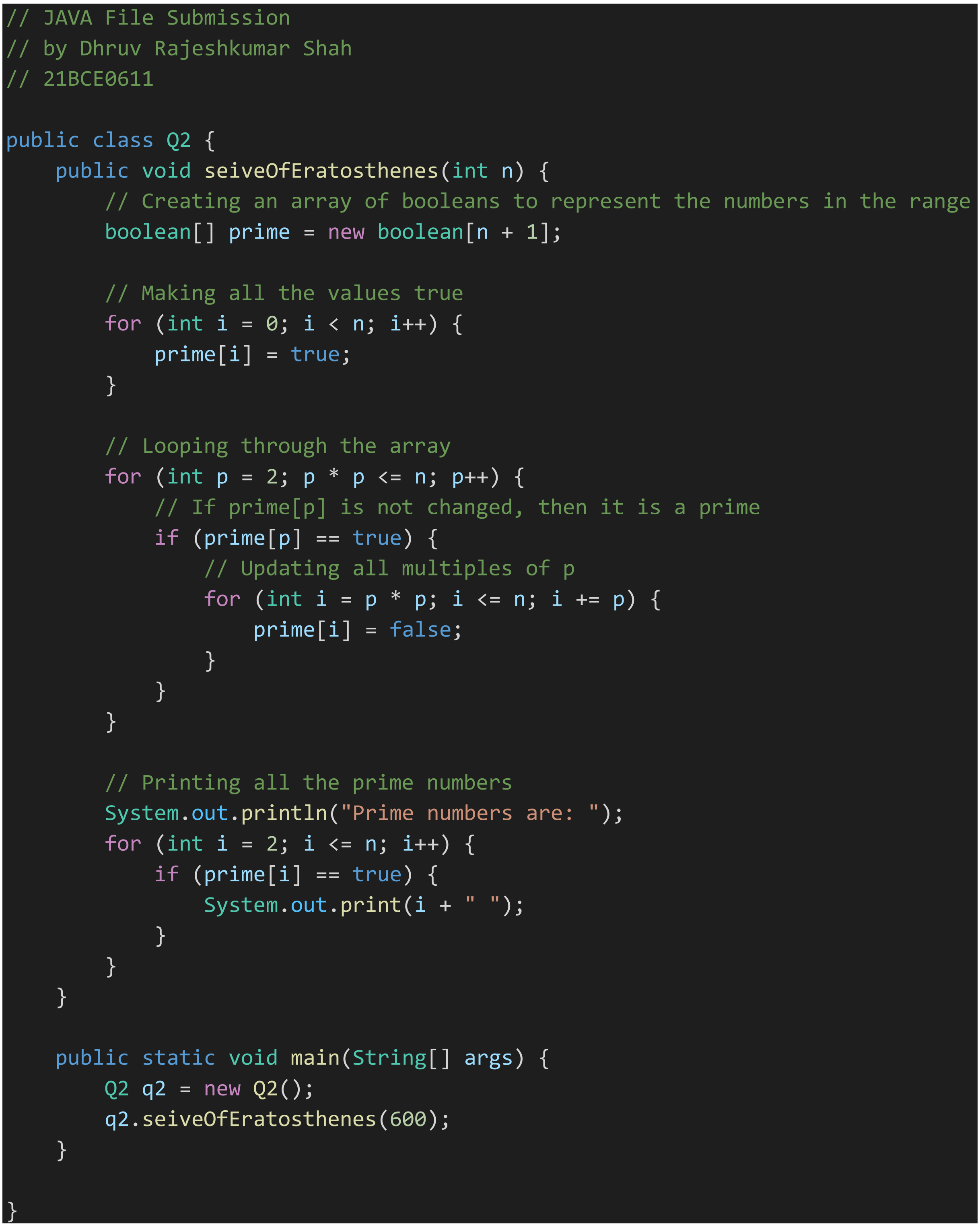
OUTPUT



```
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmission (main)
$  /usr/bin/env C:\\Program\ Files\\Java\\jdk-11.0.11\\bin\\java.exe -cp
4173f6f3d5e64de9a11\\redhat.java\\jdt_ws\\FileSubmission_461683ec\\bin Q
8
9
0
2
3
-5
2
-1
Average: 2.7142856
```

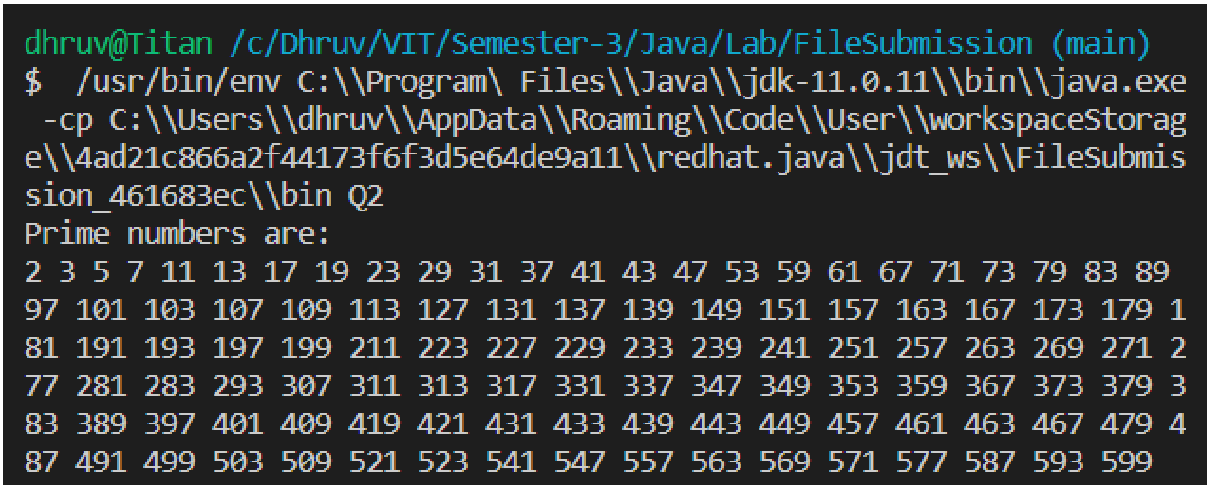
Question 2: => Write a program to print all prime numbers less than 600. Use the sieve method: => Take the first 600 integers and cross out all those that are multiples of 2, 3, 5, 7, 11 and so on, until only primes remain. Think of using a creative strategy to cross out the non prime numbers.

CODE



```java
// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

public class Q2 {
    public void seiveOfEratosthenes(int n) {
        // Creating an array of booleans to represent the numbers in the range
        boolean[] prime = new boolean[n + 1];

        // Making all the values true
        for (int i = 0; i < n; i++) {
            prime[i] = true;
        }

        // Looping through the array
        for (int p = 2; p * p <= n; p++) {
            // If prime[p] is not changed, then it is a prime
            if (prime[p] == true) {
                // Updating all multiples of p
                for (int i = p * p; i <= n; i += p) {
                    prime[i] = false;
                }
            }
        }

        // Printing all the prime numbers
        System.out.println("Prime numbers are: ");
        for (int i = 2; i <= n; i++) {
            if (prime[i] == true) {
                System.out.print(i + " ");
            }
        }
    }

    public static void main(String[] args) {
        Q2 q2 = new Q2();
        q2.seiveOfEratosthenes(600);
    }

}
```

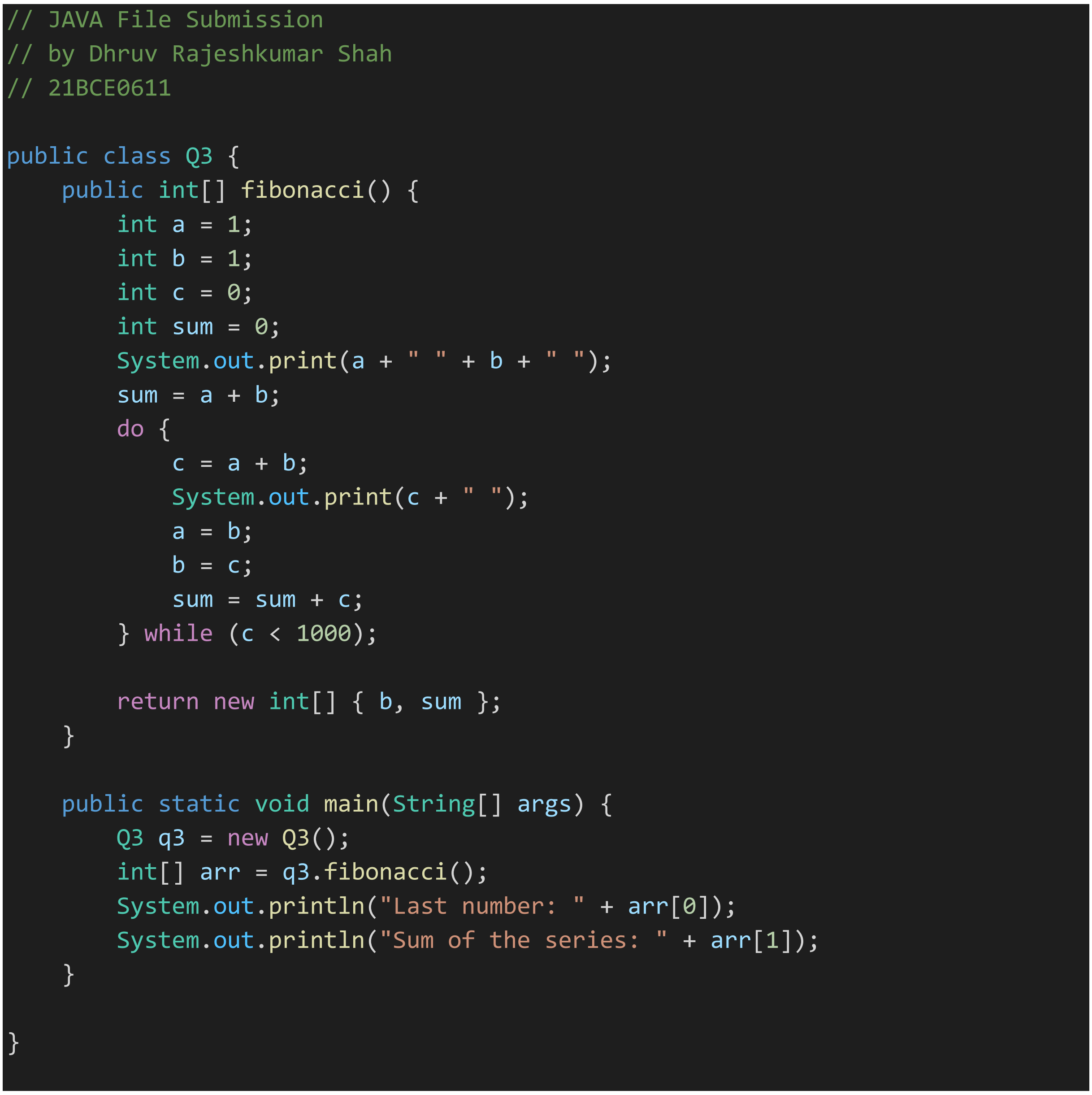
OUTPUT



```
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmission (main)
$  /usr/bin/env C:\\Program\ Files\\Java\\jdk-11.0.11\\bin\\java.exe
 -cp C:\\Users\\dhruv\\AppData\\Roaming\\Code\\User\\workspaceStorag
e\\4ad21c866a2f44173f6f3d5e64de9a11\\redhat.java\\jdt_ws\\FileSubmis
sion_461683ec\\bin Q2
Prime numbers are:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89
97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 1
81 191 193 197 199 211 223 227 229 233 239 241 251 257 263 269 271 2
77 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 3
83 389 397 401 409 419 421 431 433 439 443 449 457 461 463 467 479 4
87 491 499 503 509 521 523 541 547 557 563 569 571 577 587 593 599
```

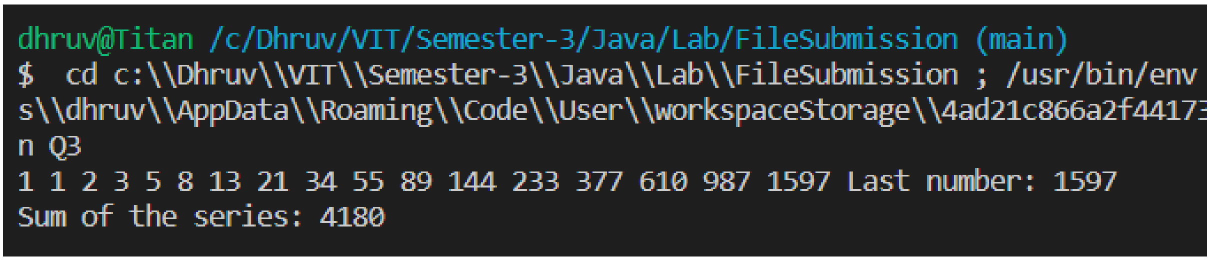
Question 3: => Find the first term greater than 1000 in the sequence: 1, 1, 2, 3, 5, 8, 13, and so on. Find the sum of all the numbers in the sequence up to that term.

CODE



```java
// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

public class Q3 {
    public int[] fibonacci() {
        int a = 1;
        int b = 1;
        int c = 0;
        int sum = 0;
        System.out.print(a + " " + b + " ");
        sum = a + b;
        do {
            c = a + b;
            System.out.print(c + " ");
            a = b;
            b = c;
            sum = sum + c;
        } while (c < 1000);

        return new int[] { b, sum };
    }

    public static void main(String[] args) {
        Q3 q3 = new Q3();
        int[] arr = q3.fibonacci();
        System.out.println("Last number: " + arr[0]);
        System.out.println("Sum of the series: " + arr[1]);
    }

}
```

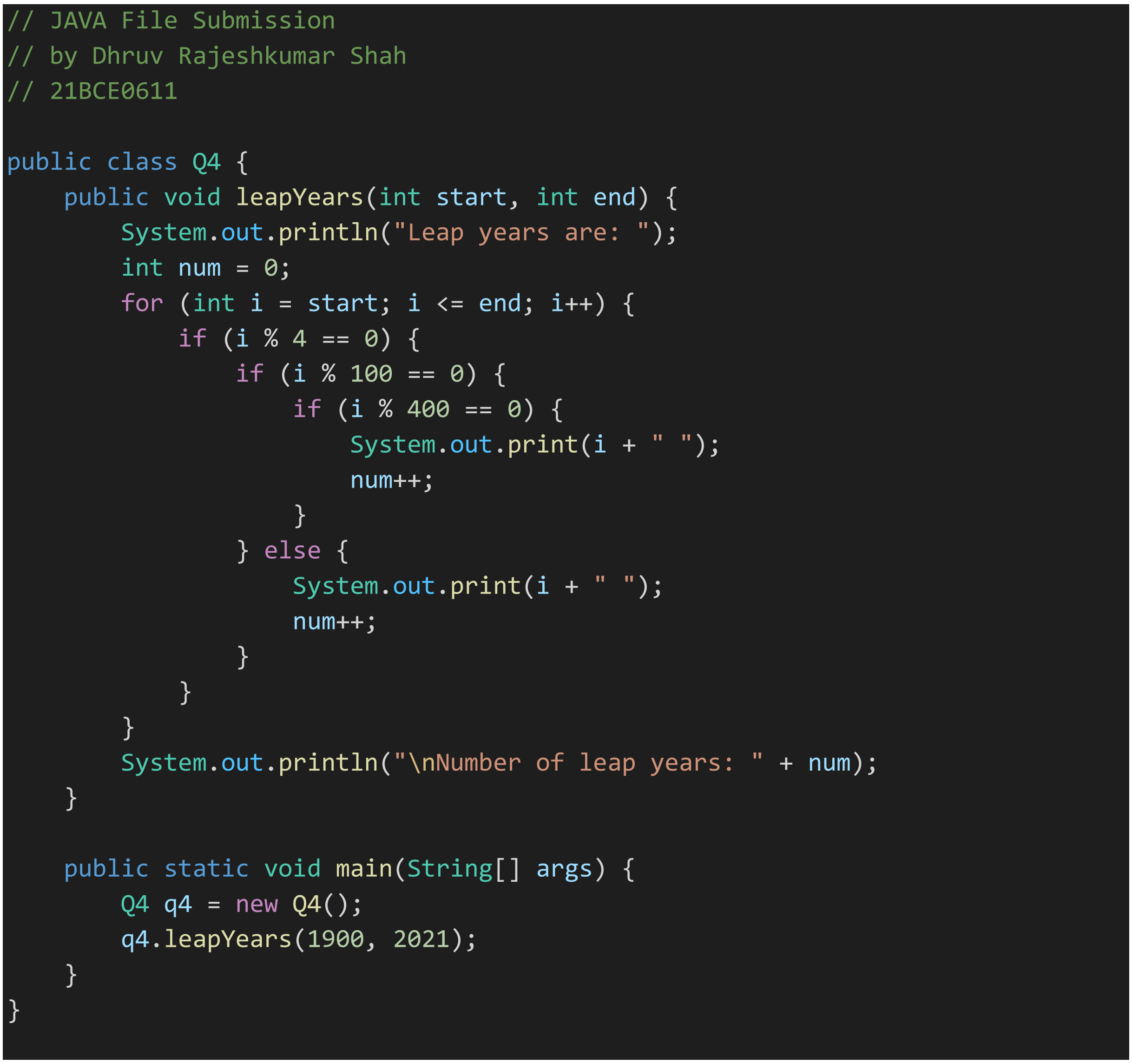
OUTPUT



```
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmission (main)
$  cd c:\\Dhruv\\VIT\\Semester-3\\Java\\Lab\\FileSubmission ; /usr/bin/env
s\\dhruv\\AppData\\Roaming\\Code\\User\\workspaceStorage\\4ad21c866a2f4417
n Q3
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 Last number: 1597
Sum of the series: 4180
```

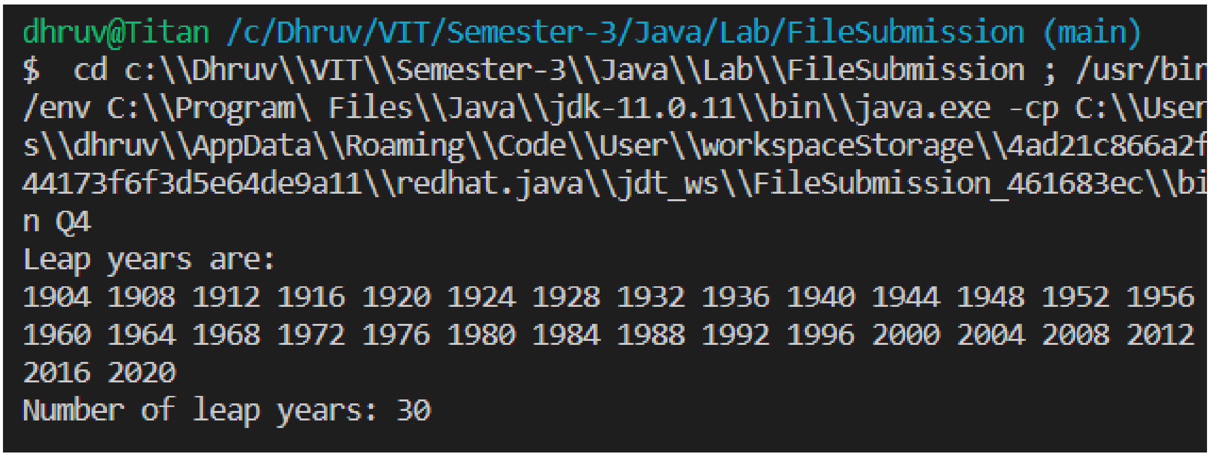
Question 4: => How many leap years occurred between 1900 and 2021 [both inclusive].

CODE



```
// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

public class Q4 {
    public void leapYears(int start, int end) {
        System.out.println("Leap years are: ");
        int num = 0;
        for (int i = start; i <= end; i++) {
            if (i % 4 == 0) {
                if (i % 100 == 0) {
                    if (i % 400 == 0) {
                        System.out.print(i + " ");
                        num++;
                    }
                } else {
                    System.out.print(i + " ");
                    num++;
                }
            }
        }
        System.out.println("\nNumber of leap years: " + num);
    }

    public static void main(String[] args) {
        Q4 q4 = new Q4();
        q4.leapYears(1900, 2021);
    }
}
```

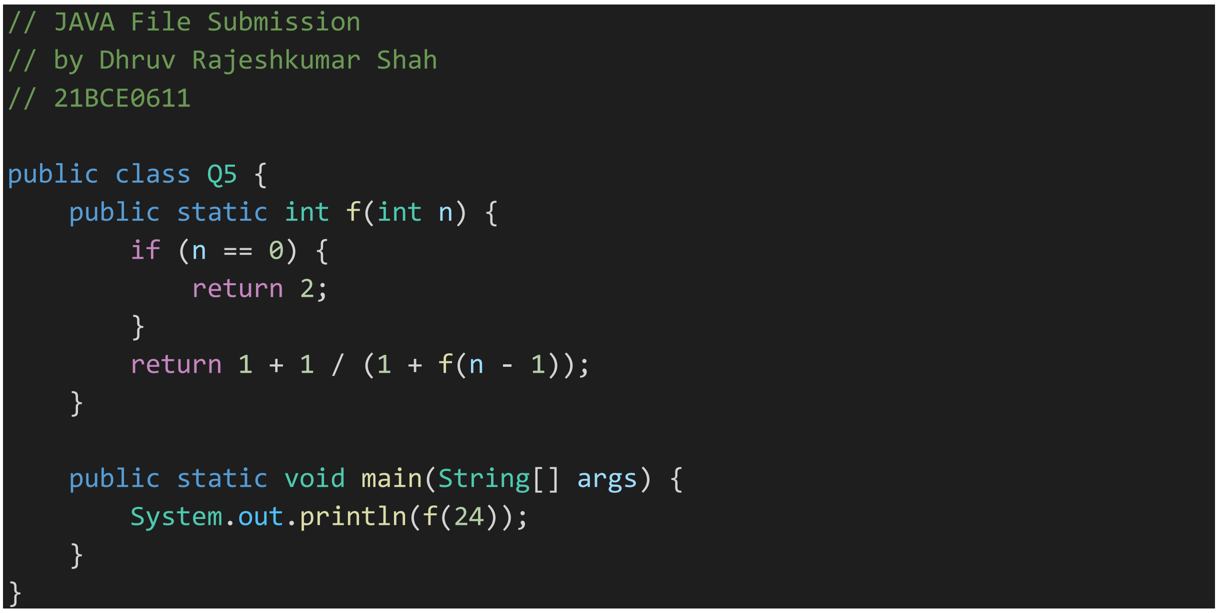
OUTPUT



```
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmission (main)
$  cd c:\\Dhruv\\VIT\\Semester-3\\Java\\Lab\\FileSubmission ; /usr/bin
/env C:\\Program\ Files\\Java\\jdk-11.0.11\\bin\\java.exe -cp C:\\User
s\\dhruv\\AppData\\Roaming\\Code\\User\\workspaceStorage\\4ad21c866a2f
44173f6f3d5e64de9a11\\redhat.java\\jdt_ws\\FileSubmission_461683ec\\bi
n Q4
Leap years are:
1904 1908 1912 1916 1920 1924 1928 1932 1936 1940 1944 1948 1952 1956
1960 1964 1968 1972 1976 1980 1984 1988 1992 1996 2000 2004 2008 2012
2016 2020
Number of leap years: 30
```

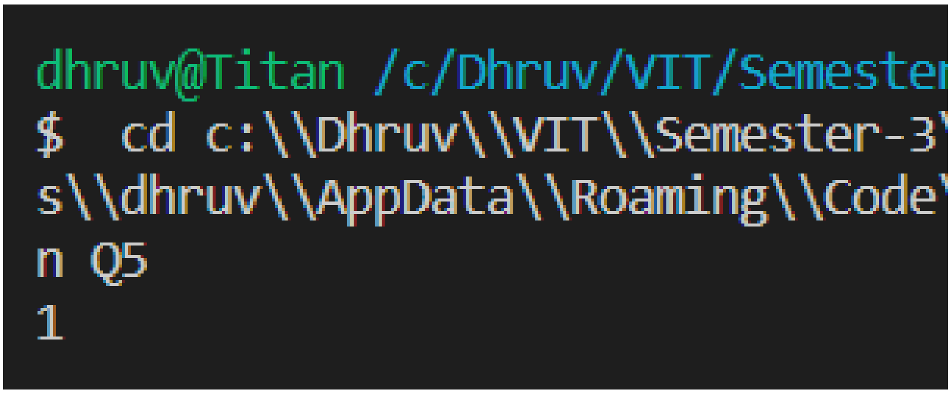
```
Question 5: => A recurrence relation is given by f(0) = 2 and
f(n+1) = 1 + 1/(1 + f(n)) (One plus (one over (one plus
f(n)))). What is f(24)?
```

CODE



```java
// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

public class Q5 {
    public static int f(int n) {
        if (n == 0) {
            return 2;
        }
        return 1 + 1 / (1 + f(n - 1));
    }

    public static void main(String[] args) {
        System.out.println(f(24));
    }
}
```

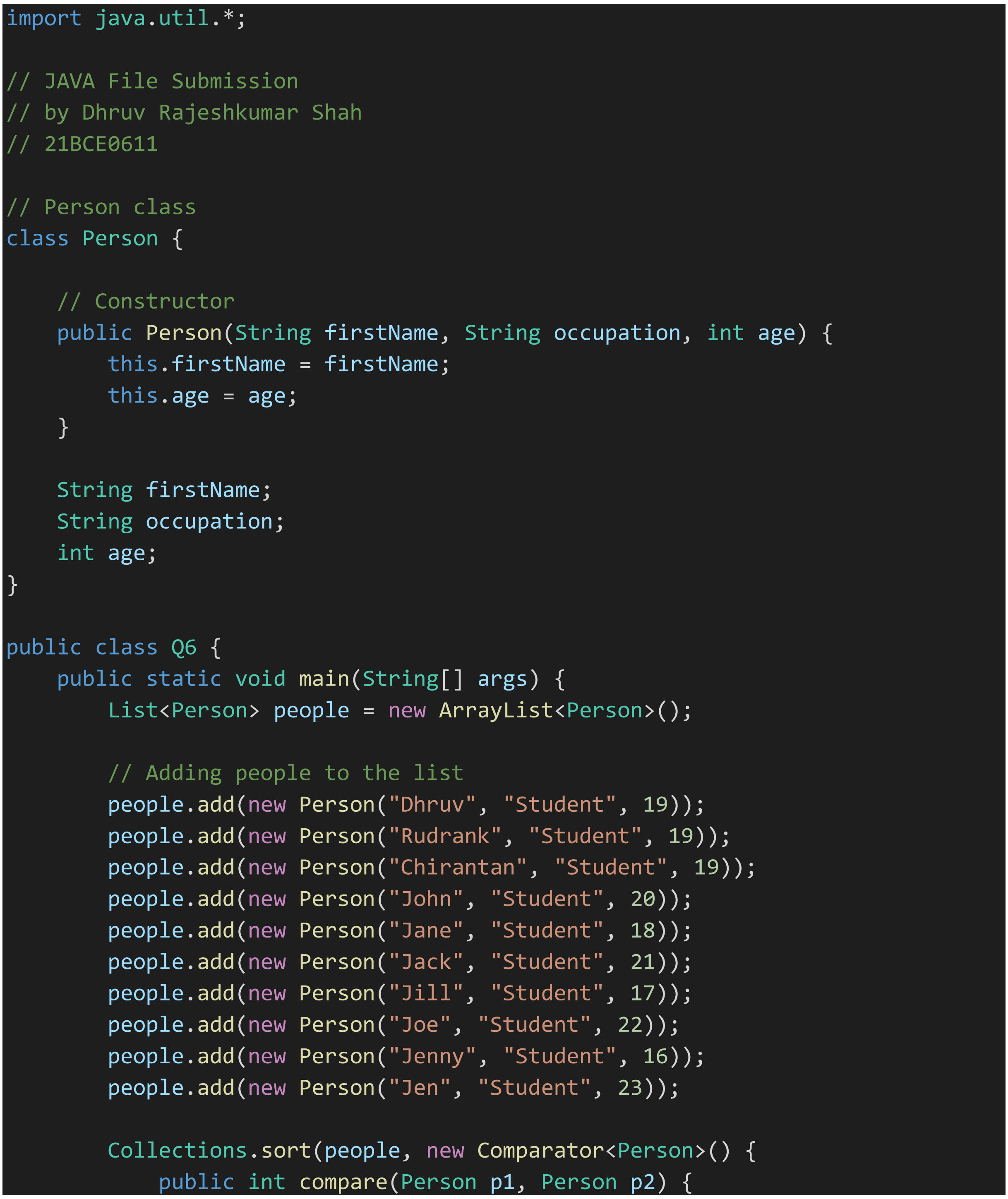
OUTPUT



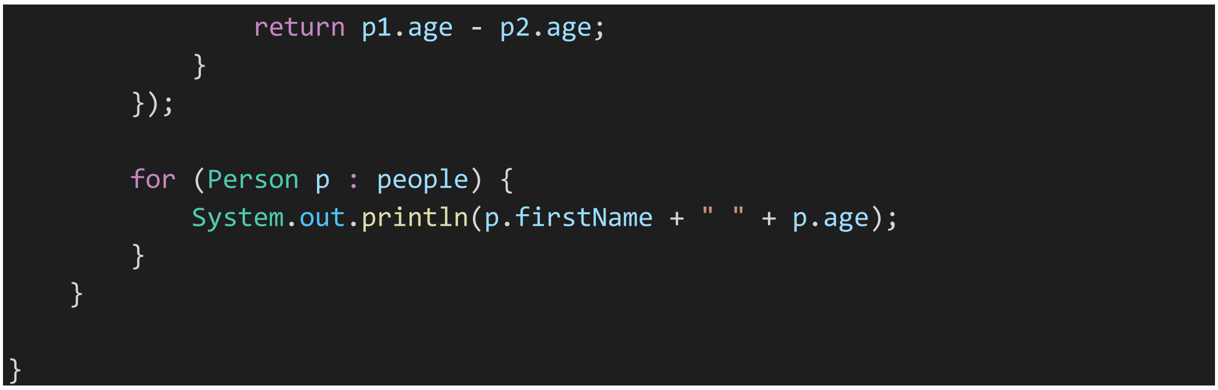
```
dhruv@Titan /c/Dhruv/VIT/Semester
$  cd c:\\Dhruv\\VIT\\Semester-3
s\\dhruv\\AppData\\Roaming\\Code
n Q5
1
```

Question 6: => Create a Person class that contains first name, age and occupation. Create an array of 10 Person objects denoted by p. Sort the array p in ascending order of age. Note that your sort method should be static, because it needs to take an overall view of all objects. Use the same sort methodology as discussed in the class in the context of ArrayList.

CODE

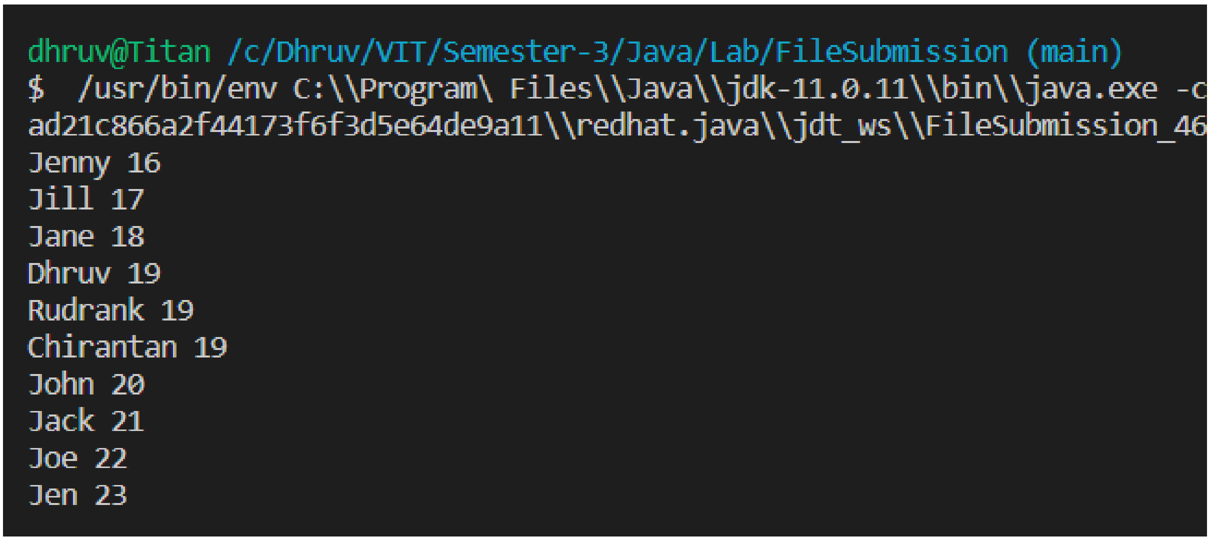


```java
import java.util.*;

// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

// Person class
class Person {

    // Constructor
    public Person(String firstName, String occupation, int age) {
        this.firstName = firstName;
        this.age = age;
    }

    String firstName;
    String occupation;
    int age;
}

public class Q6 {
    public static void main(String[] args) {
        List<Person> people = new ArrayList<Person>();

        // Adding people to the list
        people.add(new Person("Dhruv", "Student", 19));
        people.add(new Person("Rudrank", "Student", 19));
        people.add(new Person("Chirantan", "Student", 19));
        people.add(new Person("John", "Student", 20));
        people.add(new Person("Jane", "Student", 18));
        people.add(new Person("Jack", "Student", 21));
        people.add(new Person("Jill", "Student", 17));
        people.add(new Person("Joe", "Student", 22));
        people.add(new Person("Jenny", "Student", 16));
        people.add(new Person("Jen", "Student", 23));

        Collections.sort(people, new Comparator<Person>() {
            public int compare(Person p1, Person p2) {
```



```
                return p1.age - p2.age;
            }
        });

        for (Person p : people) {
            System.out.println(p.firstName + " " + p.age);
        }
    }

}
```

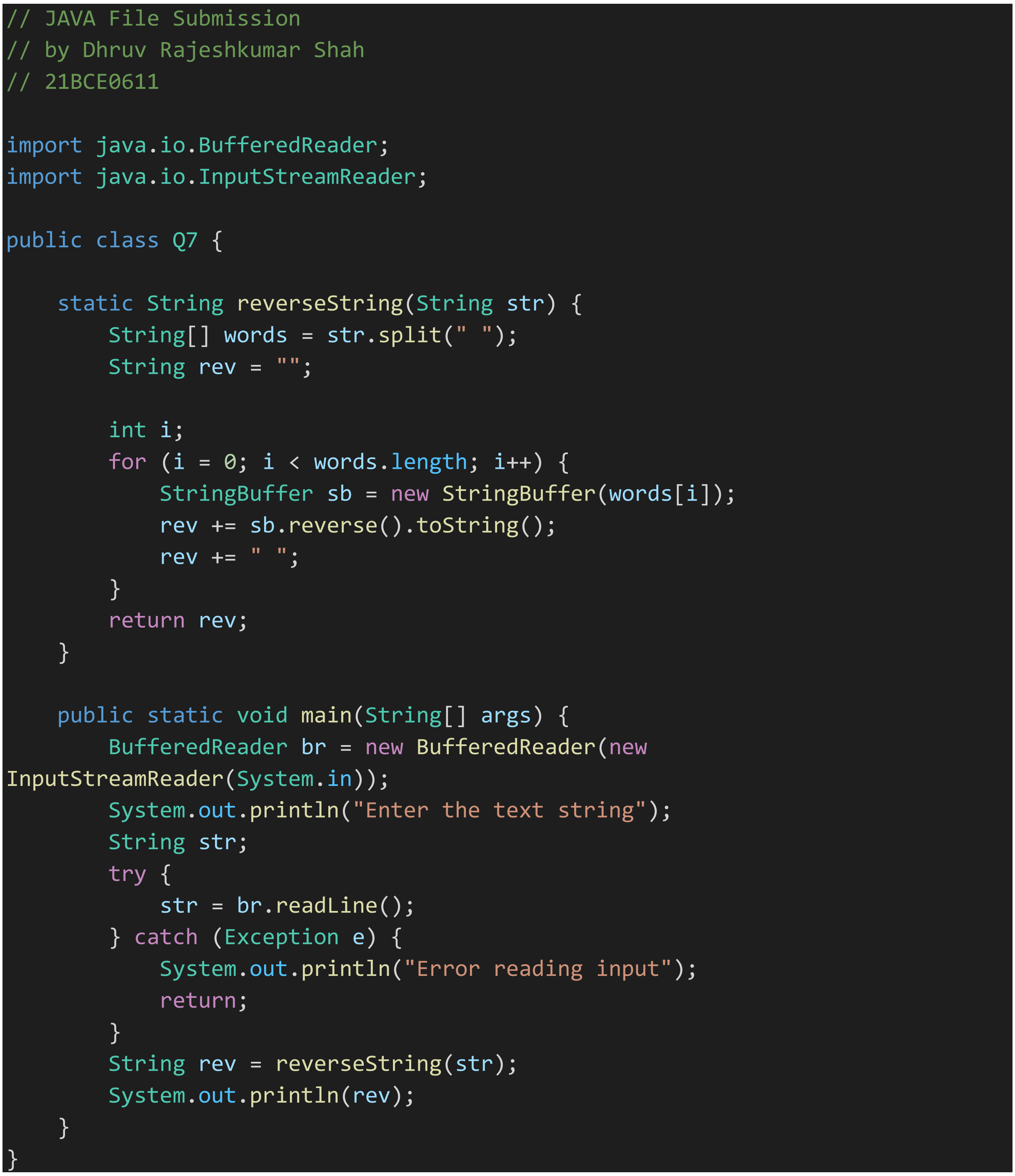
OUTPUT



```
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmission (main)
$  /usr/bin/env C:\\Program\ Files\\Java\\jdk-11.0.11\\bin\\java.exe -c
ad21c866a2f44173f6f3d5e64de9a11\\redhat.java\\jdt_ws\\FileSubmission_46
Jenny 16
Jill 17
Jane 18
Dhruv 19
Rudrank 19
Chirantan 19
John 20
Jack 21
Joe 22
Jen 23
```

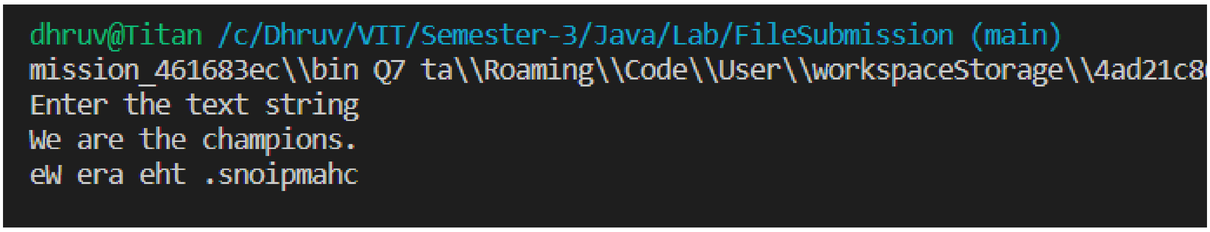
Question 7: => Using a buffered reader, read an english sentence with words separated by spaces and terminating in a newline. Write a Java method that reverses the order of letters in each word appearing in that sentence but keeps the original word order. Display the result. Assume that words are always separated by a single space and that there is no space at the beginning and at the end of the sentence. For example, “We are the champions.” is displayed as “eW era eht .snoipmahc”. Look at the HashMap lecture notes for one way of going about it.

CODE



```
// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Q7 {

    static String reverseString(String str) {
        String[] words = str.split(" ");
        String rev = "";

        int i;
        for (i = 0; i < words.length; i++) {
            StringBuffer sb = new StringBuffer(words[i]);
            rev += sb.reverse().toString();
            rev += " ";
        }
        return rev;
    }

    public static void main(String[] args) {
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        System.out.println("Enter the text string");
        String str;
        try {
            str = br.readLine();
        } catch (Exception e) {
            System.out.println("Error reading input");
            return;
        }
        String rev = reverseString(str);
        System.out.println(rev);
    }
}
```

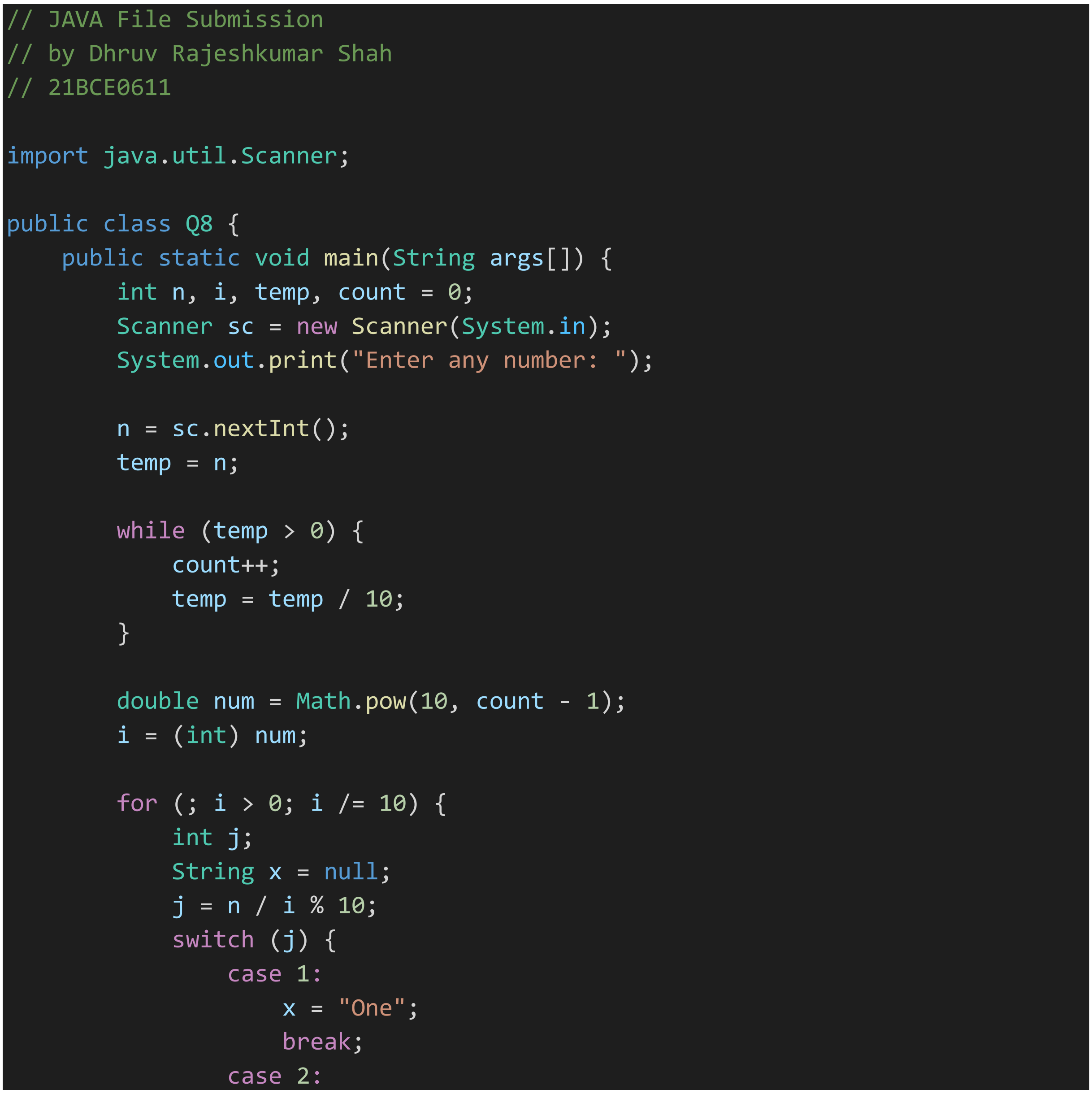
OUTPUT



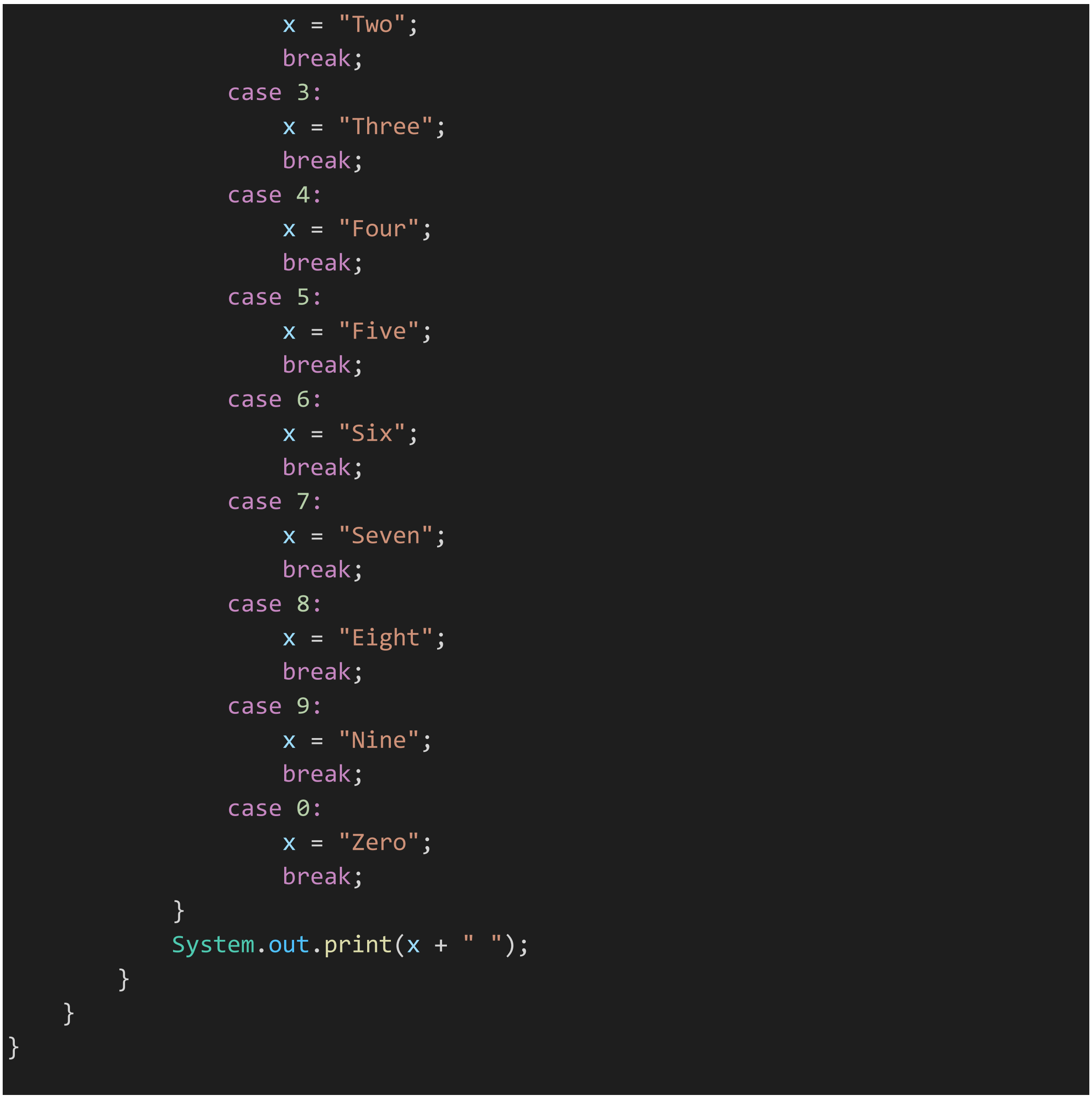
```
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmission (main)
mission_461683ec\\bin Q7 ta\\Roaming\\Code\\User\\workspaceStorage\\4ad21c8
Enter the text string
We are the champions.
eW era eht .snoipmahc
```

Question 8: => Write a Java program that breaks a number into its corresponding digits and then prints the name of the digits. For example for 1000, the program prints one zero zero zero, for 1910, the program prints one nine one zero.

CODE

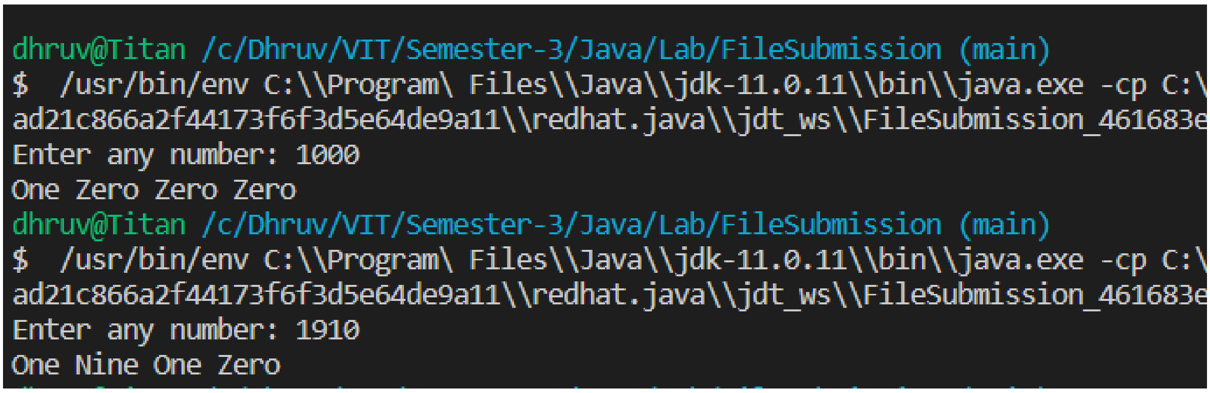


```java
// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

import java.util.Scanner;

public class Q8 {
    public static void main(String args[]) {
        int n, i, temp, count = 0;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter any number: ");

        n = sc.nextInt();
        temp = n;

        while (temp > 0) {
            count++;
            temp = temp / 10;
        }

        double num = Math.pow(10, count - 1);
        i = (int) num;

        for (; i > 0; i /= 10) {
            int j;
            String x = null;
            j = n / i % 10;
            switch (j) {
                case 1:
                    x = "One";
                    break;
                case 2:
```



```
                    x = "Two";
                    break;
                case 3:
                    x = "Three";
                    break;
                case 4:
                    x = "Four";
                    break;
                case 5:
                    x = "Five";
                    break;
                case 6:
                    x = "Six";
                    break;
                case 7:
                    x = "Seven";
                    break;
                case 8:
                    x = "Eight";
                    break;
                case 9:
                    x = "Nine";
                    break;
                case 0:
                    x = "Zero";
                    break;
            }
            System.out.print(x + " ");
        }
    }
}
```

OUTPUT



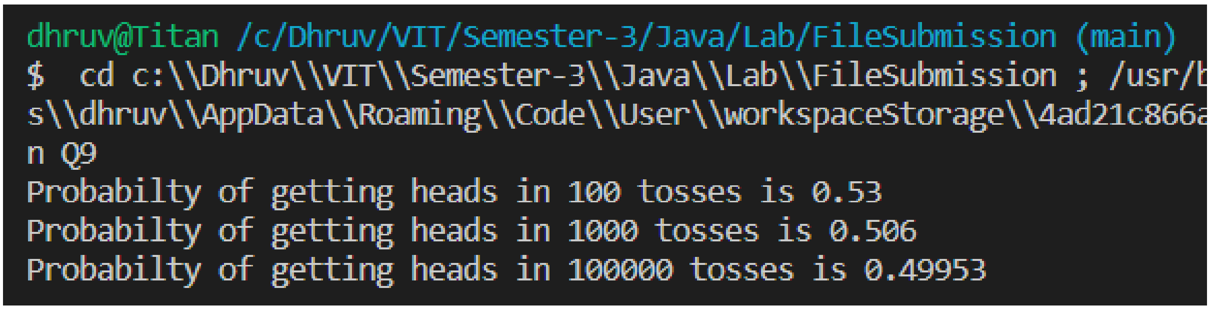
```
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmission (main)
$  /usr/bin/env C:\\Program\ Files\\Java\\jdk-11.0.11\\bin\\java.exe -cp C:\
ad21c866a2f44173f6f3d5e64de9a11\\redhat.java\\jdt_ws\\FileSubmission_461683e
Enter any number: 1000
One Zero Zero Zero
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmission (main)
$  /usr/bin/env C:\\Program\ Files\\Java\\jdk-11.0.11\\bin\\java.exe -cp C:\
ad21c866a2f44173f6f3d5e64de9a11\\redhat.java\\jdt_ws\\FileSubmission_461683e
Enter any number: 1910
One Nine One Zero
```

Question 9: => Read about java.util.Random and its nextInt(bound) method. Using this, prove that if you were to toss a coin, the probability that you would get a head is 1/2 or 0.5. The way to do this would be to assign the value 0 to the head and 1 to the tail. Sample a random value between 0 and 1, a large number of times, let's say y. Of these y trials, let's say, we got the head (0), x number of times and tail (1) y-x number of times. Then x/y should get closer to 0.5 as y grows large.

CODE

```java
// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

import java.util.Random;

public class Q9 {
    public static float getProbability(int y) {
        int x = 0;
        Random rand = new Random();
        for (int i = 0; i < y; i++) {
            int toss = rand.nextInt(2);
            if (toss == 0) {
                x++;
            }
        }
        return (float) x / y;
    }

    public static void main(String[] args) {
        int y = 100;
        System.out.println("Probabilty of getting heads in " + y + " tosses is
" + getProbability(y));
        y = 1000;
        System.out.println("Probabilty of getting heads in " + y + " tosses is
" + getProbability(y));
        y = 100000;
        System.out.println("Probabilty of getting heads in " + y + " tosses is
" + getProbability(y));

    }
}
```

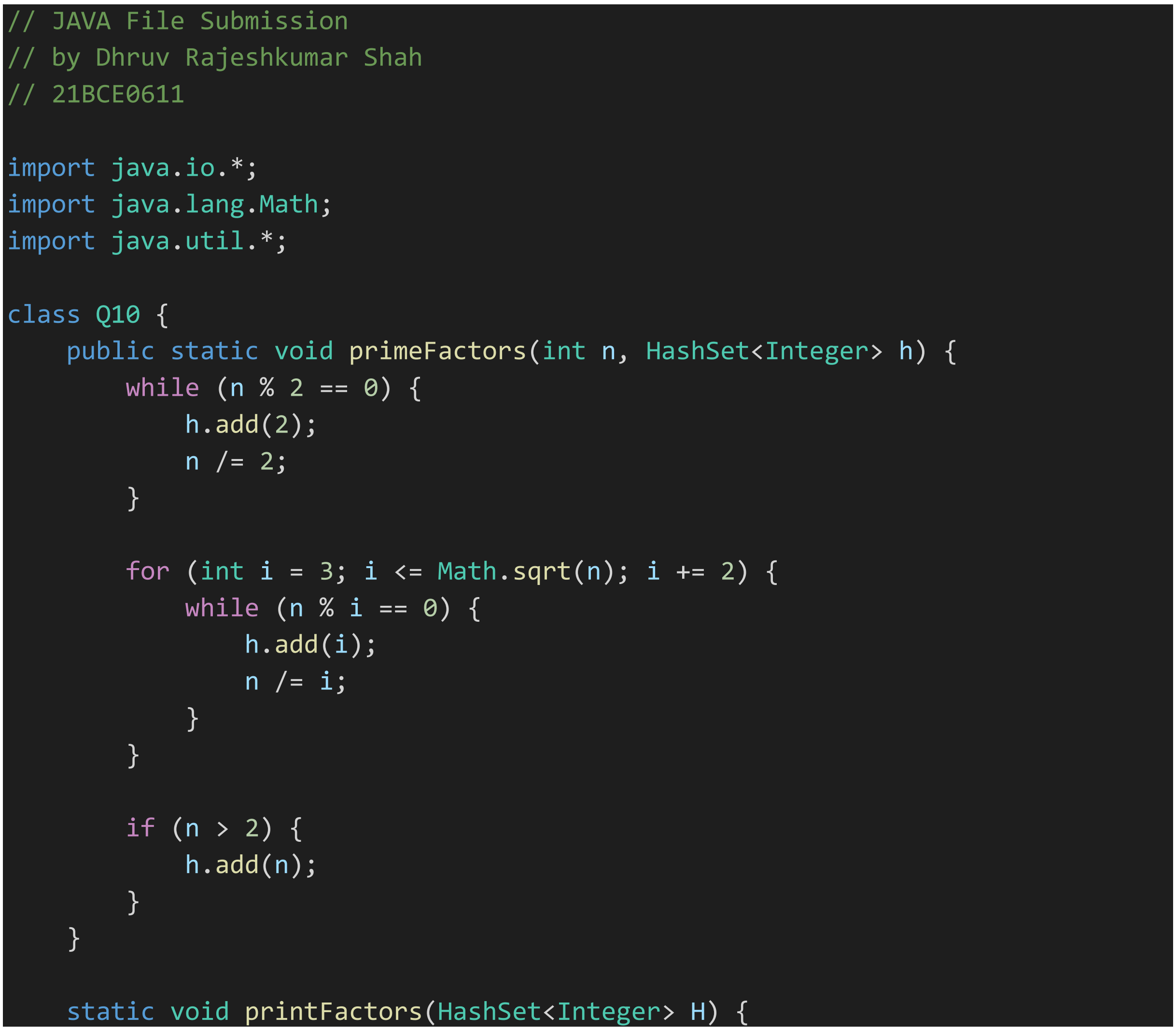
OUTPUT



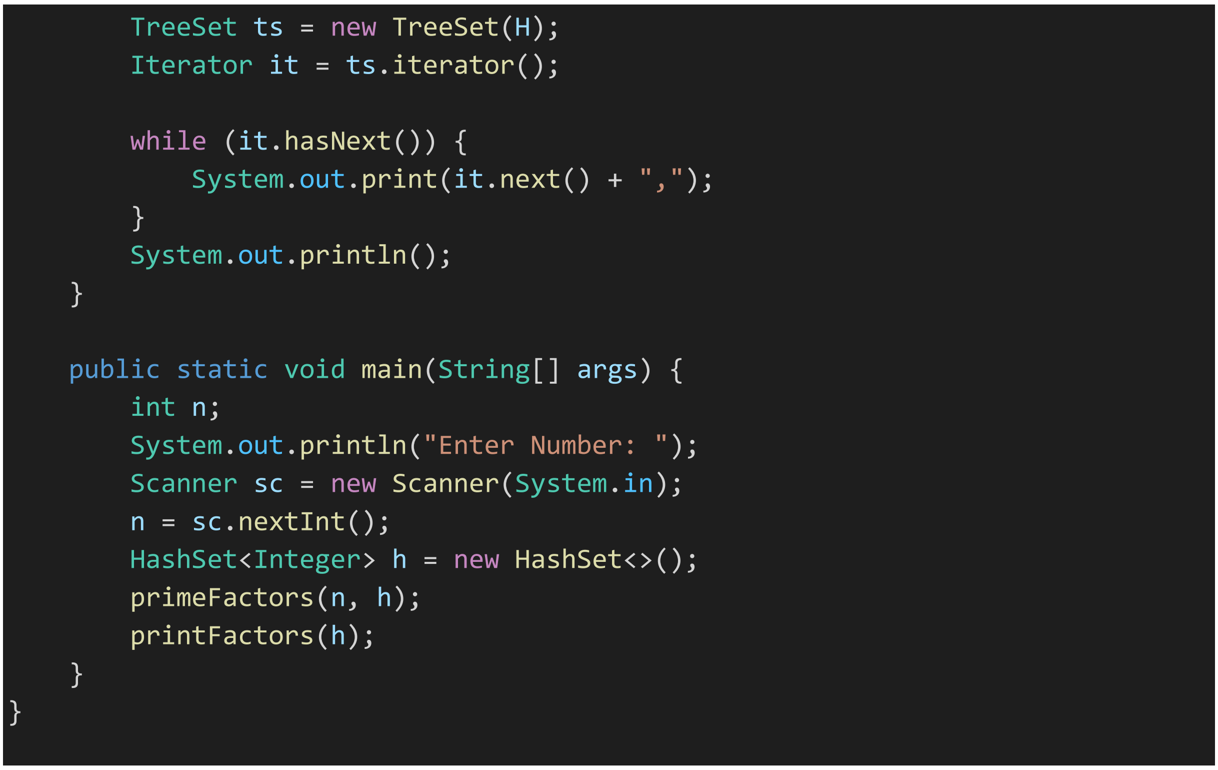
```
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmission (main)
$  cd c:\\Dhruv\\VIT\\Semester-3\\Java\\Lab\\FileSubmission ; /usr/b
s\\dhruv\\AppData\\Roaming\\Code\\User\\workspaceStorage\\4ad21c866a
n Q9
Probabilty of getting heads in 100 tosses is 0.53
Probabilty of getting heads in 1000 tosses is 0.506
Probabilty of getting heads in 100000 tosses is 0.49953
```

Question 10: => Any number can be expressed as the product of prime numbers. Given a number, such as 497553 print all prime numbers that multiply to form this number: Input 497553; Output 3,7,19,29,43. Take into account numbers like 28424 which can be written as 2^3 x 11 x 17 x 19. They must also print 2, 11, 17, 19 (no repeated 2s).

CODE

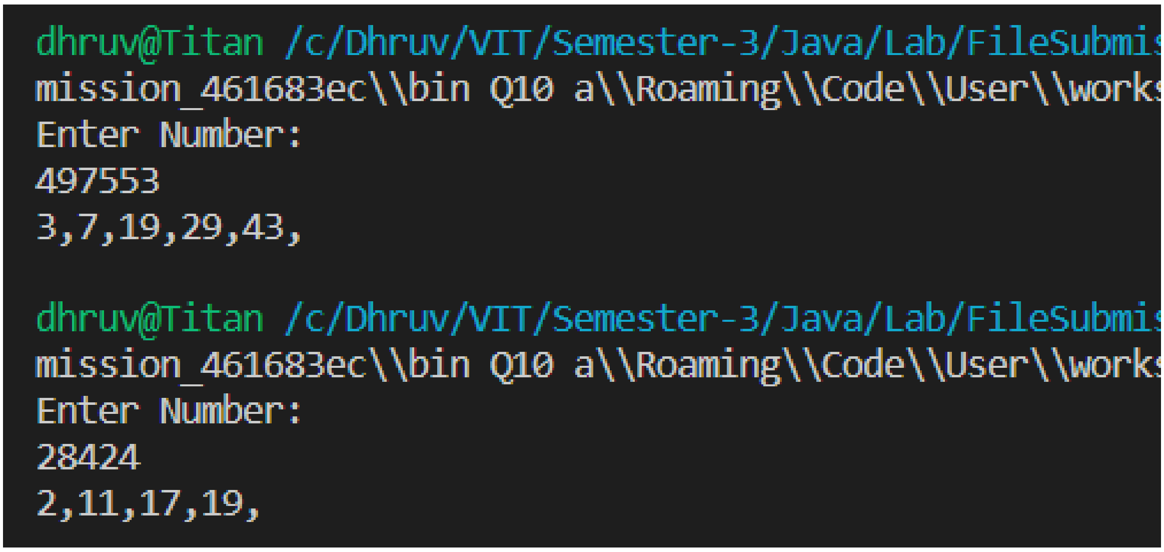


```java
// JAVA File Submission
// by Dhruv Rajeshkumar Shah
// 21BCE0611

import java.io.*;
import java.lang.Math;
import java.util.*;

class Q10 {
    public static void primeFactors(int n, HashSet<Integer> h) {
        while (n % 2 == 0) {
            h.add(2);
            n /= 2;
        }

        for (int i = 3; i <= Math.sqrt(n); i += 2) {
            while (n % i == 0) {
                h.add(i);
                n /= i;
            }
        }

        if (n > 2) {
            h.add(n);
        }
    }

    static void printFactors(HashSet<Integer> H) {
```



```
        TreeSet ts = new TreeSet(H);
        Iterator it = ts.iterator();

        while (it.hasNext()) {
            System.out.print(it.next() + ",");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        int n;
        System.out.println("Enter Number: ");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        HashSet<Integer> h = new HashSet<>();
        primeFactors(n, h);
        printFactors(h);
    }
}
```

OUTPUT



```
dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmis
mission_461683ec\\bin Q10 a\\Roaming\\Code\\User\\works
Enter Number:
497553
3,7,19,29,43,

dhruv@Titan /c/Dhruv/VIT/Semester-3/Java/Lab/FileSubmis
mission_461683ec\\bin Q10 a\\Roaming\\Code\\User\\works
Enter Number:
28424
2,11,17,19,
```